1. **Explicar o funcionamento do protocolo HTTP.**

O HTTP é o protocolo utilizado para troca de documentos de hipertexto, ou seja, é a base para a comunicação de dados da world wide web.   
Essa troca realiza-se da seguinte maneira:  
O Web Cliente envia um pedido para o Server através do browser e recebe uma resposta do Server, após este comunicar com a Database e verificar que consegue emitir uma resposta (se não o conseguir fazer, temos um erro).

1. **Explicar o significado de todos os campos da seguinte url:   
   http://www.ua.pt/deti/PageCourse.aspx?id=383&p=4&a=9**

*http* é o protocolo, *www.ua.pt* é o servidor, *deti* é o caminho, *PageCourse.aspx* é o recurso  
*?id=383&p=4&a=9* são os dados, em que *?* e *&* são os separadores

1. **Explicar o que é o HTML.**HTML é uma linguagem de marcação (Hypertext Markup Language) criada por Tim Berners-Lee utilizada para a construção/desenvolvimento de elementos que constituem os websites.
2. **Explicar o que é um marcador (tag) HTML, com exemplos.**Como o próprio nome diz, uma tag é um marcador que. na linguagem html, é responsável pela formatação do documento, indicando o tipo de elemento com que estamos a trabalhar (p.e.: a tag <p> diz-nos que estamos a trabalhar com um parágrafo).   
   Existem dois tipos de tags: as que precisam de abertura e fechamento, </ >, e as que não precisam (p.e.: <br />)
3. **Utilizando apenas HTML e CSS, implementar uma tabela com as seguintes propriedades:  
   Largura: 800px \*  
   Altura: 400px \*  
   A primeira fila deverá ser o cabeçalho da tabela com as palavras Name e Age, sendo a primeira coluna colapsada \*  
   As restantes filas deverão conter informações sobre diferentes pessoas\*  
   A tabela terá de ter uma legenda\*  
   A borda da tabela deverá ser pontilhada\*  
   O texto deverá ser centrado\*  
   A cor do corpo da página deverá ser azul.\***<body>  
   <table width=”800px” height=”400px” border=”1” style=”border-style: dotted; border-color: #488bc9; text-align: center; color: #488bc9”>  
   <caption>Insira aqui a legenda da tabela</caption>  
   <tr> <th colspan=”2”>Name</th> <th>Age</th> </tr>  
   <tr> <td>Ana</td> <td>Silva</td> <td>16 anos</td> </tr>  
   <tr> <td>João</td> <td>Amador</td> <td>26 anos</td> </tr>  
   <tr> <td>Rui</td> <td>Tavares</td> <td>32 anos</td> </tr>  
   </table>  
   </body>
4. **Explicar a estrutura e funcionamento de um formulário HTML.**Um formulário começa com a tag <form> e fecha com a tag </form>  
   Na tag <form> podemos ter ainda os atributos name, que dá um nome para o formulário para futura referência, method, que pode ser post para ocultar os dados do usuário ou get para não o fazer, e action que é onde terá o endereço da entidade processadora.  
   A seguir podemos ter várias divisões no formulário, utilizando a tag <fieldset> </fieldset>, devidamente legendadas com a tag <legend> </legend>.  
   Dentro dessas divisões opcionais podemos ter vários <input>. Cada input tem dentro de si um name, que dá um nome ao elemento que mais tarde servirá como referência; o type, que define o tipo de secção, podendo ser um radio, text, checkbox, password entre outras; um maxlength, que define o número máximo de caracteres aceites; um value, que pode ou não estar omisso ou vazio; um placeholder, que será um texto que irá pré aparecer na caixa do input.  
   Podemos também ter tags <textarea></textarea>. Dentro dessa tag também podemos ter os atributos name, placeholder e maxlength, e ainda podemos ter os atributos rows e cols que definem quantas linhas e colunas vão ser dispendidas para aquela área de texto.  
   Um formulário deve ainda ter bottons. Estes são inseridos através da tag <input> definindo o atributo type como o tipo de button desejado, p.e. type=”submit” para o botão de enviar e type=”resetbtn” para o botão de limpar formulário. Os botões podem ainda ter name e value, que no caso, esta última terá o texto que será exibido.  
   Depois de preenchido o formulário, ao clicar no botão submit, este deverá ser enviado para o link presente no action.
5. **Explicar o que é o Responsive Web Design, utilizando o Twitter Bootstrap como exemplo.**O Responsive Web Design consiste numa abordagem de web site que tem como objetivo o desenvolvimento de um site que se adapte a qualquer dispositivo, seja ele de pequenas, médias ou grandes dimensões, através da redimensionação de elementos, para, assim, oferecer uma experiência de visualização e leitura fácil.  
   O Twitter Bootstrap é um exemplo de uma framework que oferece várias componentes de HTML, CSS e JavaScript de utilidade pública para o desenvolvimento de web projetos.

1. **Desenvolver uma tabela semelhante à do exercício 5, mas utilizando o Twitter Bootstrap.**<html>  
   <head>  
   <meta name="viewport" content="width=device-width, initial-scale=1, shrink-to-fit=no">  
   <link rel="stylesheet" href="<https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/css/bootstrap.min.css>">  
   <style>  
   h3 {   
   border: 1px dotted #0094ff;  
   padding: 10px!important;  
   text-align: center;  
   vertical-align: middle;  
   color: #0094ff;  
   font-size: 0.6em;  
   margin: auto;  
   }  
   </style>  
   </head>

<body>

<div class="container">

<div class="row">

<div class="col-sm-12 col-md-6 col-lg-4">

<h3>Nome</h3>

<h3>Ana</h3>

<h3>João</h3>

<h3>Guilherme</h3>

</div>

<div class="col-sm-12 col-md-6 col-lg-4">

<h3>Sobrenome</h3>

<h3>Silva</h3>

<h3>Henriques</h3>

<h3>Loureiro</h3>

</div>

<div class="col-sm-12 col-md-6 col-lg-4">

<h3>Idade</h3>

<h3>16 anos</h3>

<h3>23 anos</h3>

<h3>34 anos</h3>

</div>

</div>

</div>

<!-- Fim da página, scripts de suporte ao Bootstrap -->

<script src="https://code.jquery.com/jquery-3.3.1.slim.min.js"></script>

<script src="https://cdnjs.cloudflare.com/ajax/libs/popper.js/1.14.7/umd/popper.min.js"></script>

<script src="https://stackpath.bootstrapcdn.com/bootstrap/4.3.1/js/bootstrap.min.js"></script>

</body>

</html>

1. **Porque motivo se deve incluir o elemento (tag) <script> no final do <body> de uma página HTML?**Quando alguma tag é encontrada, o browser vai executá-la e com a tag <script> não é diferente, ou seja, quando o browser encontrar a tag <script> ele não vai executar mais nada enquanto não terminar de executar o que está dentro dela.  
   Colocar um <script> no final do body, permite que o conteúdo antes dele já apareça para o usuário sem ter de esperar a execução do script. Isso passa a impressão de um site mais rápido. A desvantagem é que - se o seu script modifica significativamente o conteúdo e/ou sua apresentação e funcionalidade - o usuário verá uma página "estranha" e "mal formatada" antes que o script a "corrija".
2. **Desenvolva uma calculadora utilizando uma função Javascript, acedendo aos elementos do DOM.**

<!DOCTYPE html>

<html>

<head>

<meta charset="utf-8" />

<title></title>

</head>

<body>

<div id="calculadora">

<input id="op1" />

<select id="operador">

<option value="+">+</option>

<option value="-">-</option>

<option value="\*">\*</option>

<option value="/">/</option>

</select>

<input id="op2" />

<button onclick="calculadora()">Res</button>

</div>

<script>

function calculadora() {

var op1 = document.getElementById("op1").value;

var op2 = document.getElementById("op2").value;

var op = document.getElementById("operador").value;

var res = 0;

switch (op) {

case "+":

res = parseFloat(op1) + parseFloat(op2);

break;

case "-":

res = op1 - op2;

break;

case "\*":

res = op1 \* op2;

break;

case "/":

if (op2 == 0) {

alert("não podemos dividir por zero");

} else {

res = op1 / op2;

break;

}

}

alert("Resultado = " + res);

}

</script>

</body>

</html>